Physics-based Human Motion Estimation and Synthesis from Videos

Kevin Xie[1,2], Tingwu Wang[1,2], Umar Iqbal[2]
Yunrong Guo[2], Sanja Fidler[1,2], Florian Shkurti[1]
[1]University of Toronto and Vector Institute, [2]Nvidia
kevincxie@cs.toronto.edu

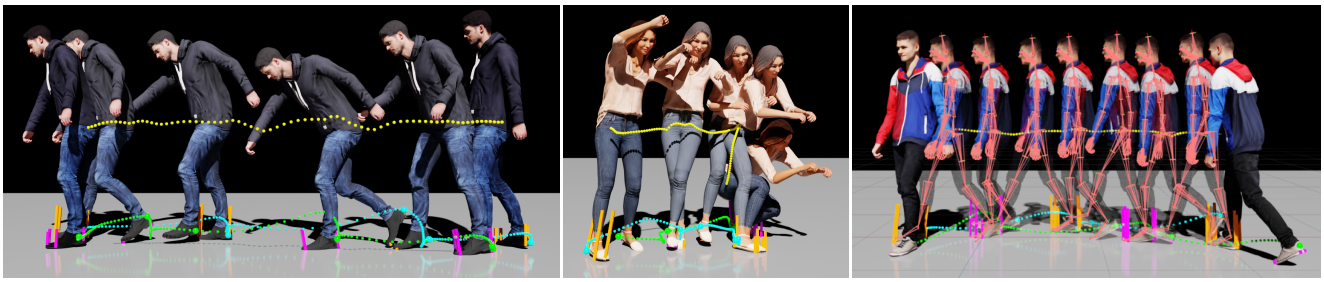

Figure 1: We propose a framework to estimate physically correct motions from noisy pose estimations from video. This allows us to train a motion synthesis network directly on video data, removing the need for mocap data used in prior work.

Abstract

Human motion synthesis is an important problem with applications in graphics, gaming and simulation environments for robotics. Existing methods require accurate motion capture data for training, which is costly to obtain. Instead, we propose a framework for training generative models of physically plausible human motion directly from monocular RGB videos, which are much more widely available. At the core of our method is a novel optimization formulation that corrects imperfect image-based pose estimations by enforcing physics constraints and reasons about contacts in a differentiable way. This optimization yields corrected 3D poses and motions, as well as their corresponding contact forces. Results show that our physically-corrected motions significantly outperform prior work on pose estimation. We can then use these to train a generative model to synthesize future motion. We demonstrate both qualitatively and quantitatively significantly improved motion estimation, synthesis quality and physical plausibility achieved by our method on the large scale Human3.6m dataset [12] as compared to prior kinematic and physics-based methods. By enabling learning of motion synthesis from video, our method paves the way for large-scale, realistic and diverse motion synthesis.

1. Introduction

Given videos of human motion, how can we infer the 3D trajectory of the body's structure and use it to generate new, plausible movements that obey physics constraints? Addressing the intricacies of this question opens up an array of possibilities for high-fidelity character animation and motion synthesis, informed by real-world motion. This would benefit games, pedestrian simulation [55] in testing environments for self-driving cars, realistic long-horizon predictions for model-based control and reinforcement learning, as well as physics-based visual tracking.

The vast majority of existing approaches in learning-based human motion synthesis [1, 59, 26, 21, 5] rely on large-scale motion capture observations, such as AMASS [32], which are typically costly and time-consuming to acquire, logistically challenging, and most often limited to recordings in indoor environments. These factors form a bottleneck that hinders the collection of high-quality human motion data, particularly in settings where there is interaction among multiple people or interaction with a number of stationary and moving objects in the scene. The recorded motions typically also lack realism and diversity as they are acquired by acting out a set of pre-defined motions. In addition to this issue, many time-series models trained on motion capture data make predictions that are oblivious to the physics constraints of motion and contact, often leading to inaccurate, jerky, and implausible motion.

In this paper we entirely forego reliance on motion capture and aim to train physically plausible human motion synthesis directly from monocular RGB videos. We propose a framework that refines noisy image-based pose estimates by enforcing physics constraints through contact invariant optimization [37, 36], including computation of con-

tact forces. We then use the results of the refinement to train a time-series generative model that synthesizes both future motion and contact forces. Our contributions are:

- We introduce a smooth contact loss function to perform physics-based refinement of pose estimates, eschewing the need for separately trained contact detectors or nonlinear programming solvers.

- We demonstrate that when visual pose estimation is combined with our physics-based optimization, even without access to motion capture datasets, it is sufficient to train motion synthesis models that approach the quality of motion capture prediction models.

We validate our method on the Human3.6m dataset [12], and demonstrate both qualitatively and quantitatively the improved motion synthesis quality and physical plausibility achieved by our method, compared to prior work on learning-based motion prediction models, such as PhysCap [44], HMR [16], HMMR [58], and VIBE [18].

2. Related Works

We organize the rich existing literature on motion synthesis across two axes: (a) kinematic vs. physics-based methods, and (b) imitation learning vs. model-based control and reinforcement learning. Table 1 provides a summary of the most relevant works.

2.1. Kinematic Motion Synthesis

Kinematic motion synthesis models make predictions without necessarily satisfying physics constraints. *Non-parametric methods* in this category attempt to blend motion clips and concatenate them into a coherent trajectory. Examples of this type of work include motion matching [5] and the use of motion graphs [19, 42] and motion fields [21] in character animation.

Parametric kinematic methods, on the other hand, rely on pose predictions made by a time-series generative model, typically a neural network. After training, the example motions are not used for prediction anymore, in contrast to non-parametric approaches. To maintain consistency in the predicted motion many papers make use of motion generation via recurrent neural networks (RNN) [7, 33, 54, 61, 8, 47], variational autoencoders for time-series data [26, 10, 56], autoregressive models [11, 46], transformers [23], or by explicitly maintaining a memory bank of past motions.

2.2. Physics-Based Motion Synthesis

Physics-based animation methods make motion predictions that satisfy the body dynamics and are informed by physics constraints [2], often including contacts, which adds to the realism of the generated movement. Seminal work in *contact-invariant optimization* [37, 36] introduced soft inverse dynamics constraints to optimize center-of-mass trajectories as well as contact forces without requiring explicit planning of contact locations. In [35] and in [40], it was shown that this framework could be sped up and also be used in a setting where target velocities are selected interactively.

	Input Modality	Physics	Functionality
DLow [56]	mocap		synthesis
RFC [57]	mocap	$\sim$	synthesis
MOJO [59]	mocap		synthesis
PhysCap [44]	video	✓	pose est.
Rempe *et al*. [43]	video	✓	pose est.
Ours	video	✓	pose est., synthesis

Table 1: Comparison of features of different related works. RFC uses a physics simulator but does not use proper contact dynamics.

In addition to imposing soft physics constraints, recent reinforcement learning controllers have been used for motion synthesis with *hard physical constraints* [29, 28]. These approaches leverage model-based sampling planning to generate physically correct motions which corrects pose estimation errors and model mismatch. Injecting hard physics constraints for dynamics and contact has been a fruitful approach in trajectory optimization for humanoid robots [6], which typically makes use of nonlinear programming solvers and mixed integer-quadratic programs. Incorporating example motions and training data into these optimization frameworks, however, is challenging. So is generating diverse motions. Moreover, execution times of these frameworks are typically not suitable for real-time operation.

To balance the fidelity of dynamics with the cost of computation time, simplified physics models such as centroidal dynamics models, or models that enforce soft-dynamics constraints, have been commonly used in literature. For example, [52, 20] use centroidal dynamics to fine-tune character motion from physically incorrect motion templates.

Model-free reinforcement learning approaches, which do not assume known or learned dynamics, are gaining popularity due to their flexibility, efficiency in high-dimensional motion synthesis that tracks realistic reference motion. In DeepMimic [41], model-free controllers are trained to output torques to follow the reference motion. DeepMimic is able to physically correctly reproduce a large variety of motion skills. However it takes hours or days just to reproduce one motion. Since then, efforts have been made to extend model-free controllers. In [53, 51], by improving the capacity of neural networks, controllers can now master all the skills in a large motion dataset without having to retrain for each motion as in DeepMimic.

2.3. Kinematic and Physics-Based Pose Estimation

Purely kinematic approaches for 3D pose and shape [60, 59] estimation from video, such as HMMR [17], VIBE [18], and XNect [34], predict past and future motion, without incorporating physics constraints. Physics constraints, however, can act as a regularizer, adding temporal consistency to the estimated 3D motion. Both motion capture and human video data have been used as observations in pose estimation, with the latter modality leading to an ill-posed problem. For example, PhysCap [44] achieves physically plausible real-time human motion estimation in 3D from videos, including modeling of contacts and prediction of their locations, which leads to minimal foot-to-floor penetration. [43] also models hard contact constraints, which cannot be changed after detection. Physics-based visual tracking [50] provides additional examples of work in this area, including ones that handle contacts [24, 4] as hard constraints during trajectory optimization, as well as entire meshes [30].

Our main difference from these works is that by using our proposed soft contact penalty, contact events can form dynamically and softly during optimization. Our method does not need separate contact labelling, and instead of a complex alternating optimization with discrete steps to relabel contacts, it optimizes in two contiguous passes with an off-the-shelf unconstrained LBFGS optimizer.

3. Method

An overview of our proposed framework for learning motion synthesis from videos can be seen Fig 2. It consists of four steps: **1)** Given an unlabeled video, we estimate the positions of 2D and 3D body joints at each video frame using a monocular pose estimation model [14]. **2)** We then transform the 3D body joints at each frame to relative body-part rotations of the parametric body model, SMPL [31], using inverse-kinematics [15, 22]. **3)** We then refine the initial motion estimates using our proposed physics-based optimization which results in physically plausible and temporal coherent motion for the entire video. **4)** We process all available videos with aforementioned steps, and subsequently use the resulting motions to train our motion synthesis model. In the following we detail each step.

3.1. 3D Pose Estimation

Given an unlabeled RGB video, we estimate the 3D body pose from each frame using a monocular pose estimation model. For this, we chose the method of [14, 13] as it provides 3D body pose in absolute camera coordinates. We follow [14] and use HRNet-w32 [48] as the backbone, and train it on Humans3.6M [12], 3DPW [49] and MSCOCO [25] datasets. At each frame, it provides the 3D pose $\mathbf{p^{pe}} \in \mathbb{R}^{J\times 3}$ and 2D pose $\mathbf{p^{pe,2d}} \in \mathbb{R}^{J\times 2}$, where the 3D pose $\mathbf{x}$ is estimated up to a scaling factor. The global

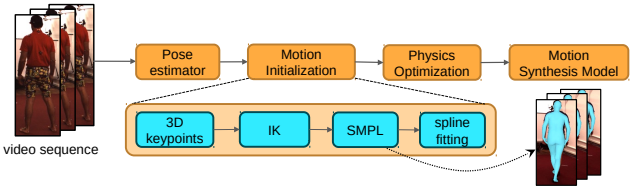

Figure 2: **Overview of our framework.** A video sequence is processed by a per-frame CNN pose estimator. The 3d and 2d keypoint detections are passed to an inverse kinematics step that forms an initial estimate of the SMPL body model motion using 3D keypoints. We then optimize this initialization with our physics loss and use the produced motions in place of motion capture to train motion synthesis models.

scale of the person is approximated using the mean bone length, which obviously is sub-optimal and leads to physically implausible results, *e.g.* feet penetrating the ground-plane. Since the pose of each frame is estimated independently, we found that the resulting poses contain a large amount of jitter both spatially as well as in terms of scale.

3.2. Motion Representation and Initialization

The 3D positions are not the most optimal representation for effectively modelling the spatial and temporal inter-/intra-part correlations, as slight changes in the depth of the person under the same pose will lead to significantly different 3D positions for all body parts. Hence, we convert the 3D positions to local body-part rotations using the analytical inverse-kinematics method of [22] using swing-twist decomposition. Similar to [22], we use the parametric body model SMPL [31] to kinematically represent the body motion. SMPL consists of a linear function that takes the pose parameters $\theta \in \mathbb{R}^{24\times 3}$ and shape parameters $\beta \in \mathbb{R}^{10}$ as input and produces an articulated triangle mesh $\mathbf{M} \in \mathbb{R}^{6980\times 3}$ containing 6890 vertices. Like in SMPL, we parameterize joint rotations with the exponential map representation. The method of [22] uses a trained model to predict the twist component for all body joints. In this work, we initially set the twist to zero, and optimize it as part of the physics optimization as we will explain later in this section.

Given the 3D body pose in all video frames $t = [0, T]$ represented as local rotations θ_t, we first remove the the high-frequency noise by smoothing the motion using a Butterworth low-pass filter. The exponential map rotation suffers from singularities at 2π rotations, hence we model global root rotation by separating out global yaw rotation and representing it with per-frame rotation offsets. Specifically[1],

$$\theta_t^{root} = \left(\sum_{\tau=0}^{t} \Delta\theta_\tau^{root,yaw}\right) * \theta_t^{root,xy}$$

[1]We apply all mathematical operations on joint rotations including the optimization using quaternion, but leave the conversion for brevity.

Variable	Description
β	Body shape parameter for SMPL model. It is static over time.
p_t^{root}	Root position.
$\Delta\theta_t^{root,yaw}$	Delta axis angle rotation only along the z gravity direction.
θ_t^{joints}	Axis angle local rotation of joints root for each spline knot.
$\theta_t^{root,xy}$	XY rotation of root as unnormalized xy quaternion.
f_t^c	Scaled contact forces at the n_c contact sites.
Δt_i	Delta time to next spline knot.

Table 2: Overview of variables that are directly optimized, their symbol and description. For all of the variables that depend on time, we are actually optimizing the parameters of their respective splines (including tangent values).

We apply the same smoothing procedure to the global root positions p_t^{root} as well. Gathering these, we represent the overall motion in generalized coordinates $q_t = \{p_t^{root}, \theta_t\}$.

Although this motion sequence can then be optimized directly, we further model the motion with cubic splines to constrain our motions to be smooth and reduce the dimensionality of our optimization variables. Specifically, we use cubic Hermite splines where the node positions of the spline are initialized to temporally evenly spaced frames covering the whole motion (effectively subsampling it by a factor of 8) and the tangents are initialized according to the rule for Catmull-Rom splines. To compute the full motion sequence, we simply query the spline at the sampling times of the original motion sequence. We also optimize the timings Δt_i between the spline knots but found its inclusion to have negligible impact on the final results.

3.3. Motion Optimization

In the motion optimization step, we refine the motion by jointly optimizing the body shape β and global character poses $\{q_t\}_{t=1:T}$ to match both the pose estimator detections as well as a full-body physics loss term that uses a smooth contact penalty [38]. Note that we optimize only one set of shape parameters β for the entire sequence, as the identity of the person does not change within a sequence. This stage also optimizes corresponding ground contact forces f_t^c which we parameterize with splines just as we do for the pose. The total loss function to be optimized in our method combines a physics loss with a pose estimation loss and a smoothness regularization.

$$L_{total} = L_{pose} + L_{physics} + L_{smooth}$$

We detail each part below. We evaluate the losses at evenly spaced discrete time points in the motion and average over the entire sequence.

3.3.1 Physics Loss

We now detail the computation of our differentiable physics loss function, given a motion and associated contact forces.

Assume that a temporally evenly spaced sequence of motion frames $\{q_t\}_{1:T}$ and contact forces $\{f_t^c\}_{1:T}$ are given, where q_t and f_t^c represent the generalized coordinates and global contact forces of the body at time t. The loss function consists of three main parts:

$$L_{physics}(q_t, f_t^c) = L_{dynamics} + L_{contact} + L_{penetration} \tag{1}$$

The dynamics loss penalizes impossible forces. Rigid body dynamics satisfy the Newton-Euler equations which admits a unique inverse dynamics function mapping motions to the required generalized forces that would give rise to them.

$$f_t^r(q(\cdot)) = M\ddot{q}_t + C\dot{q}_t + g \tag{2}$$

The inverse dynamics computation involving mass matrix M, centrifugal and coriolis forces $C\dot{q}_t$ and gravity g can be efficiently computed using the Recursive Newton Euler algorithm which exploits the sparsity structure induced by the kinematic tree and we use finite difference approximations for the time derivatives of $q(t)$. For a thorough tutorial on rigid body dynamics, one can refer to [27]. Using f_t^r we can calculate the dynamics loss by comparing it to the actual forces on the character.

$$L_{dynamics} = w_{dynamics}||f_t^r - Bf_t^a - J^T f_t^c||^2 \tag{3}$$

Here J^T maps all the contact forces from the contact points onto the full space and similarly B maps joint actuation f_t^a forces to the full space. Instead of leaving f_t^a as yet another optimization variable, the optimal value of Bf_t^a can be easily chosen by assuming no limits on actuation force. Practically this means that only residual forces on the root (and other unactuated joints) will be penalized and otherwise it is assumed that any extra acceleration is due to actuation. Magnitude of joint actuation is implicitly limited by penalizing acceleration of 3d joint positions and rotations, described later.

The humanoid character model is approximated with boxes, cylinders and spheres and differentiably scaled as a function of the skeleton. We detail this in the supplementary. Full-body inertia is accurately accounted for in the inverse dynamics loss and does not make use of centroidal approximations as in prior work [36]. Contact forces are assumed to be exerted only by the feet at 4 different contact points (which we will refer to as end effectors) per foot that lie on the corners of the box approximation to the feet as in [44] and [43], although more contact locations could be readily added to the current framework.

The contact cost penalizes violation of Signorini's con-

ditions for contact:

$$L_{contact} = \sum_{i}^{n_c} c_{t,i} \left(w_e ||e_{t,i}||^2 + w_{\dot{e}} ||\dot{e}_{t,i}||^2 \right) \quad (4)$$

Here, $e_{t,i} \in \mathbb{R}^3$ is the minimum displacement between the i$^{\text{th}}$ end effector position and the contact surface and its time derivative $\dot{e}_{t,i}$ is also included to prevent slip. They are penalized in proportion to the contact variable $c_{t,i}$ related to the contact force. The contact variable represents the degree to which a contact is present at that time step. It ranges from 0 to 1 and is obtained through a soft step function of the contact force magnitude as:

$$c_{t,i} = \frac{1}{2}(\tanh(k_1 ||f^c_{t,i}|| - k_2) + 1) \quad (5)$$

The contact variable is a monotonically increasing function of the contact force. Furthermore, it saturates for large values of f^c. This can be seen as a soft-relaxation of the hard step function in the complementarity condition. Intuitively, optimality is reached by bringing the contact force to zero and/or the contact distance to zero. Slipping and rolling contacts are not considered in this work.

Without further restrictions the contact objective only penalizes violation of the Signorini conditions when it specifically chooses to apply contact force. As such the method can generate motions with penetrating objects without contact force. To avoid this, a separate term is used to explicitly penalize interpenetration:

$$L_{penetration} = w_{pen} \sum_{i}^{n_c} \max(\{d_{t,i} + k_{margin}, 0\})^2 \quad (6)$$

Here, $d_{t,i}$ is the signed distance of the contact surface at the i$^{\text{th}}$ end effector which is negative if it is penetrating.

3.3.2 Pose Estimation Loss

The pose fitting loss L_{pose} we use is common in human shape estimation [3]. L_{pose} is evaluated per frame and summed. It measures the motion error in terms of local 3d keypoints deviation, global camera projected 2d keypoint deviations, log probability of the motion under a pose prior and deviation of the SMPL body shape from the mean body shape.

We also use a kinematic acceleration penalty to ensure our motions are smooth.

$$L_{smooth} = \frac{1}{n_{joints}} (w_{\ddot{\theta}} ||\ddot{\theta}_t||^2 + w_{\ddot{p}} ||\ddot{p}_t||^2) \quad (7)$$

Here $\ddot{p}_t$ is the global linear acceleration of the joints.

All of our loss terms have tuned weights which are detailed in the supplementary along with additional loss details. Although our method is not overly sensitive to this tuning, it is important to have a good balance between weighing $L_{dynamics}$ and $L_{contact}$. Outside of that, the balance between $L_{physics}$ and L_{pose} was loosely tuned such that L_{pose} does not deviate much from a purely kinematic optimization.

3.3.3 Implementation Details

We implement our full pipeline in PyTorch and use an off-the-shelf implementation of the LBFGS optimizer [39] with a history size of 100, base step size of 1.0 and Armijo-Wolfe line search. The optimization is run in 2 stages totalling 750 iterations. First 250 iterations of kinematic optimization is performed where the only difference is that $L_{physics}$ loss is disabled, then 500 iterations of physics optimization is performed with $L_{physics}$ enabled. The LBFGS memory is cleared between the 2 stages.

3.4. Generative Model

Once our motion has been optimized we can use it like a standard motion capture dataset.

In particular, we demonstrate that it can be used to train motion synthesis model that are typically only trained on mocap datasets. We follow prior work in generative human motion synthesis and adopt the state of the art Diversifying Latent Flows (DLow) method [56]. DLow uses a standard recurrent conditional VAE (CVAE) with a GRU encoder, auto-regressive decoder architecture to predict future motion given a short clip of past motion as context. Additionally it uses a learned post-hoc sampling strategy that optimizes directly for both best-of-1 accuracy and diversity of a finite number set of future motion predictions.

DLow takes as input and produces as output a sequence of root relative 3d keypoint positions and root velocities.

4. Experimental Results

In this section we evaluate our method and compare to previous work. We split our evaluations into the two stages of our pipeline. We first provide experimental details of our evaluation setting 4.1. Next, we evaluate our physics refinement step for pose estimation and compare against state of the art physics-based approach PhysCap [44] and pose estimators HMR [16], HMMR [17], and VIBE [18]. Finally, we demonstrate the benefits of using our physics optimization correction in terms of downstream performance on motion synthesis.

4.1. Dataset and Experimental Setting

We use the large scale Human3.6M dataset for our evaluations (and additional comparison to [43] on HumanEva [45] is provided in the supplementary). Motions were recorded from 4 cameras and a motion capture system was used to produce accurate annotations for the character.

	HMR [16]	HMMR [17]	PhysCap [44]	Ours (kin)	Ours (dyn)	VIBE*
no Procrustes MPJPE (↓)	78.9	79.4	97.4	73.6	**68.1**	65.6
global root position (↓)	204.2	231.1	182.6	148.2	**85.1**	-
e_{smooth} (↓)	11.2	6.8	7.2	5.42	**4.0**	-
σ_{smooth} (↓)	12.7	5.9	6.9	**1.06**	1.3	-

Table 3: Comparison of pose estimation accuracy and quality metrics for our method with physics (dyn) and without physics (kin) along with competitive pose estimator baselines. All errors are measured in millimeters. VIBE [18] is a strong oracle method that uses the large-scale AMASS [32] motion capture dataset for training. Note that as PhysCap [44] and the other baselines operate at 25fps, we downsample our 50fps motion for making a direct comparison.

We use subjects 9 and 11 which form the standard validation set and use the same motions as PhysCap [44]. Specifically, these motions do not include interactions with the chair object or lying/sitting motions. They are: *directions, discussions, greeting, posing, purchases, taking photos, waiting, walking, walking dog and walking together*.

4.2. Physics-corrected Pose Estimation

Through our evaluation we want to answer the following questions: **1)** Does our proposed physics loss improve the accuracy of pose estimation?, **2)** Does it improve the physical plausibility of pose estimation?, and **3)** How does our method compare against other physics/temporal pose estimation methods?

As we do not have access to the DeepCap dataset [9], we evaluate our method on the large scale Human3.6m dataset. We split motions into even chunks such that they are below 2000 frames (40 seconds). Most motions can be processed in one or two chunks, but a few motions require three chunks. Optimization completes in 3-4 minutes for a chunk of length 40 seconds.

Baselines. To address the first 2 points, we introduce a kinematic optimization baseline, which is equivalent to our method in all aspects, except that $L_{physics}$ is not included in the total loss for optimization (and consequently the end effector forces are also not included in the optimization variables). We also compare against HMMR [17], which is a kinematic 3D mesh and pose prediction model from videos of human motions in the wild. We further compare to its predecessor, HMR [16], which performs a similar function given a single RGB image, as opposed to a video. Our third baseline is PhysCap [44], a physics-based 3D pose prediction model from monocular video that includes contact modeling and minimizes foot-to-floor penetration, unlike other similar methods. We also compare against VIBE [18], a strong oracle that predicts both pose and body shape, but which has been trained on the large-scale AMASS [32] motion capture dataset. Similarly to HMR and HMMR, VIBE relies on an adversarial objectives that discriminates between mocap motion and predicted motion.

Evaluation metrics. We adopt evaluation metrics outlined in PhysCap [44]. Following standard practice, we measure mean per joint position error (MPJPE) on the 15 joint reduced skeleton and the mean global root position error. The e_{smooth} loss is also introduced in PhysCap and we report it as well. It measures the difference in 3d keypoint velocity magnitude between the ground truth motion and the predicted motion which illustrates the amount of jittering present in the motion and is computed as follows:

$$\hat{Jit} = ||\hat{p}_t - \hat{p}_{t-1}|| \tag{8}$$

$$Jit^{GT} = ||p_t^{GT} - p_{t-1}^{GT}|| \tag{9}$$

$$e_{smooth} = \sum_{t}^{T} \sum_{joints} ||p_t^{GT} - p_{t-1}^{GT}|| \tag{10}$$

Pose estimators that do not make use of physics losses often violate the conditions of static contact. We create metrics based on the foot joint that directly aim to measure this. Contact condition violation occurs in two ways which we design metrics for to test. To evaluate foot floating artefacts, we compare foot global z position error ($e_{foot,z}$) on ground truth:

$$e_{foot,z} = mean(|\hat{p}_{foot,z} - p_{foot,z}^{GT}|) \tag{11}$$

To evaluate foot sliding artefacts, we compare foot global xy velocity error ($e_{foot,vxy}$) with respect to ground truth.

$$e_{foot,vxy} = mean(||\Delta_t \hat{p}_{foot,xy} - \Delta_t p_{foot,xy}^{GT}||) \tag{12}$$

Results. We detail our pose estimation accuracy results in Table 3. Our method greatly outperforms PhysCap [44] on root-aligned mean joint position error without procrustes alignment. In fact, our method approaches learning-based video pose estimation methods that leverage the large scale AMASS motion capture datasets [32] to form a motion prior. Our kinematic motion baseline is competitive with HMR [16] and HMMR [17] on its own, demonstrating the power of optimization-based pose estimation.

Furthermore, we greatly improve in terms of global root position estimation as well. We attribute this to the fact that we optimize motion and body shape jointly along with our

	$e_{foot,vxy}$ (↓)	$e_{foot,z}$ (↓)
Ours (kin)	4.65	95.7
Ours (dyn)	**2.71**	**18.9**

Table 4: Ablation comparison of contact-sensitive metrics, foot tangential velocity error ($e_{foot,vxy}$) and foot global height error ($e_{foot,z}$) with and without physics loss.

contact-aware physics loss. Thereby the movement of 2d joint detections over time can help estimation of the bone lengths as can be seen Fig 3, instead of taking the initial average bone lengths without further refinement as done in PhysCap [44]. This shows that relying on temporally learned pose estimators alone to recover global scale and bone lengths is suboptimal. PhysCap does not have a direct mechanism to allow these bone lengths to be optimized with respect to contact aware losses as they optimize in separate iterative stages. However, we have a single differentiable objective that is jointly optimized with all variables including shape parameters.

The addition of the physics loss makes a noticable improvement in terms of MPJPE and a very large improvement on global root position and e_{smooth}. The large improvement in the joint speed error e_{smooth} is immediately visible in videos of the two approaches which are included in supplementary. For our kinematic baseline, without guidance on when contacts are formed and broken, feet joints of the character can often slide side to side during contact and without enforcement of the Newton Euler equations, the root of the character can freely move without limitations and often slide side to side during fast walking phases.

The major contributor to the difference in global root position error is due to depth ambiguity. Whereas, the kinematic baseline can only form a rough approximation to depth using body priors and motion cues, our physics loss enforces contact with the ground plane directly, greatly improving depth estimation. We further find the benefits of including the physics loss on the physical plausibility of contacts through our custom metrics outlined in Table 4. Specifically, the physics loss reduces foot tangent velocity error by more than 40% and height error by 80%.

Qualitative results. Measuring the quality of motion capture is difficult and quantitative metrics do not always paint the full picture. We include qualitative examples of our output as composited renders. We also show representative failure cases from the most inaccurate frames of our predictions in Fig. 5.

Many of the largest error cases occur near crouching motions. Here we are mainly limited by our geometric character approximation. The box geometry of our character does not capture the true underlying foot geometry. Our model is unable to represent significant foot flexion. However, we note that the character pose is still stable and the contact of our end effectors still engaged with the ground plane.

We also note that estimated contacts are also realistic. In fact, even though we do not use a contact detection network, we are still able to estimate realistic forces from only video

Figure 3: **Optimization result on video.** Here we show a photo snapping motion produced by our framework, video frames from the input motion are included below.

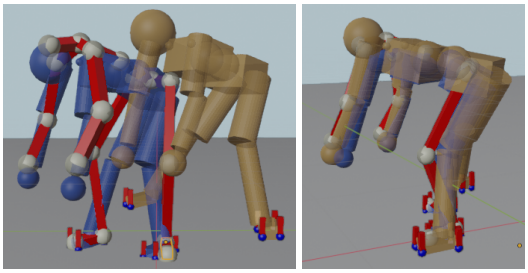

Figure 4: **Pose estimation result.** In light orange is the motion initialization for our optimization, in blue is the final output of our method overlayed on the red skeleton which is ground truth joints. In the camera view on the right, the initial pose looks plausible, but is refined drastically as the body shape is optimized by our method as seen on the side view shown on the left.

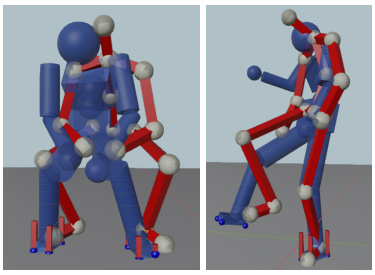

Figure 5: **Failure cases.** Even when the mocap reconstruction error is quite high, the motion output of our method still evaluates consistently low under our physics loss and visually looks physically plausible. These failure cases are selected from the worst performing frames in terms of mocap reconstruction.

input. Figure 3 qualitatively demonstrates estimated ground contact forces during a typical walking gait.

4.3. Motion Synthesis from Video

Here we show results for our combined framework that trains a motion generative model from video. Through our evaluation we want to answer the following questions: **1)** When trained with only our pose estimation-generated data, can we learn high quality motion synthesis models?, **2)** How much does our physics loss in the pose estimation step improve the performance of the down-stream motion synthesis model?

To address these questions, we train the same DLow [56] model with 3 different training datasets. DLow(GT) is the oracle model that trains with actual mocap data. DLow(PE-dyn) is our proposed method that uses the physics corrected pose estimation results from the previous stage. DLow(PE-kin) is a baseline that uses the kinematically-optimized pose estimation results from the previous stage and is used to ablate the benefit of using physics loss. We also include the results of the standard VAEs trained in the different fashions without DLow sampling.

We keep experimental settings identical between the 3 and the only varying factor is the input training data. Ultimately we evaluate the trained motion synthesis models on the ground truth validation set. As the Human3.6M [12] dataset contains multi-view cameras for each motion, we only make use of videos from the first camera to generate our datasets which simulates a monocular RGB video setting. We follow a similar evaluation protocol to DLow [56] and compare against it.

Additionally, as we limit our pose estimator to the validation set of Human3.6m, we only train the motion synthesis model with motions from the two characters, (S9 and S11). Therefore we split the motions from S9 and S11 evenly into a training and evaluation set for the motion synthesis model. Specifically, every motion named *'[Action] 1'* is used for validation leaving, one other motion of the same action type in the training set.

Apart from this, we use the exact same experimental settings as in DLow [56]. Given a context of 0.5 seconds, DLow predicts the future 2 seconds of the motion. All motions are sampled at the original 50FPS of the mocap. We use DLow with 10 output motion modes.

Evaluation metrics. We report standard metrics used in motion synthesis. The two distinct objectives of motion synthesis is to generate diverse, yet accurate motions. Accuracy is measured on the 15 joint skeleton model, average distance error (ADE) measures average root-aligned joint position error averaged over the predicted future motion sequence and final distance error (FDE) is the same, but measured only at the final frame of the predicted motion, which emphasizes longer term accuracy. Both metrics are in meters. Diversity is measured by average pairwise distance (APD). Given the set of samples produced by the motion synthesizer this gives the average L2 distance between all pairs of motion samples.

	Diversity (↑)	ADE (↓)	FDE (↓)
DLow(PE-kin)	10.53	0.590	0.698
DLow(PE-dyn)	**10.96**	**0.573**	**0.685**
DLow(GT)*	12.22	0.490	0.617
cVAE(PE-kin)	7.419	0.639	0.756
cVAE(PE-dyn)	**7.413**	**0.612**	**0.738**
cVAE(GT)*	6.801	0.5617	0.706
ERD(GT)*	0	0.722	0.969

Table 5: Comparison of motion synthesis diversity and accuracy between motion synthesis models with different training data. Note that the errors are measured in meters as we stick to the convention in motion synthesis works. The (GT)* denotes that the method was trained with ground truth mocap data, not estimated from video and should be understood to be an oracle baseline. PE-dyn is using our physics corrected pose estimation dataset and PE-kin is ablating away the physics loss in the physics correction.

Results. We tabulate the evaluation of our models in Table 5. The cVAE is the VAE that forms the backbone for the DLow method. As expected we do not match the quality of the oracle model which uses ground truth motion capture data. However, we are very competitive with this oracle. DLow trained using our physics corrected input (PE-dyn) is only worse in average joint distance by 16.9%, in final distance error by 11.0% and in average motion diversity by 10.3%. For both the DLow model and the cVAE model, adding physics loss to the correction step for generating training data consistently improves all evaluated metrics.

Qualitative results. Please see the supplementary for videos and visualizations of the motions produced from our trained motion model.

5. Conclusion

In this paper, we introduced a new framework for training motion synthesis models from raw video pose estimations without making use of motion capture data. Our framework refines noisy pose estimates by enforcing physics constraints through contact invariant optimization, including computation of contact forces. We then train a time-series generative model on the refined poses, synthesizing both future motion and contact forces. Our results demonstrated significant performance boosts in both, pose-estimation via our physics-based refinement, and motion synthesis results from video. We hope that our work will lead to more scaleable human motion synthesis by leveraging large online video resources.

References

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *ICCV*, 2019. 1

[2] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. Drecon: Data-driven responsive control of physics-based characters. *ToG*, 38(6), 2019. 2

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 5

[4] M. A. Brubaker, L. Sigal, and D. J. Fleet. Estimating contact dynamics. In *ICCV*, 2009. 3

[5] Simon Clavet. Motion matching and the road to next-gen animation. *Proceedings of GDC*, 2016. 1, 2

[6] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti. Variable horizon mpc with swing foot dynamics for bipedal walking control. *IEEE Robotics and Automation Letters*, 2021. 2

[7] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354, 2015. 2

[8] P. Ghosh, J. Song, E. Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. *3DV*, 2017. 2

[9] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*, 2020. 6

[10] I. Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and T. Komura. A recurrent variational autoencoder for human motion synthesis. In *BMVC*, 2017. 2

[11] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ToG*, 39(6), 2020. 2

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 36(7):1325–1339, 2014. 1, 2, 3, 8

[13] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via 2.5D latent heatmap regression. In *ECCV*, 2018. 3

[14] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *CVPR*, 2020. 3

[15] Umar Iqbal, Kevin Xie, Yunrong Guo, Jan Kautz, and Pavlo Molchanov. KAMA: 3d keypoint aware body mesh articulation. In *ArXiv*, 2021. 3

[16] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 2, 5, 6

[17] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, 2019. 3, 5, 6

[18] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 2, 3, 5, 6

[19] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ToG*, 21(3):473–482, 2002. 2

[20] Taesoo Kwon, Yoonsang Lee, and Michiel Van De Panne. Fast and flexible multilegged locomotion using learned centroidal dynamics. *ToG*, 39(4):46–1, 2020. 2

[21] Yongjoon Lee, Kevin Wampler, Gilbert Bernstein, Jovan Popović, and Zoran Popović. Motion fields for interactive character locomotion. *ToG*, 29(6), 2010. 1, 2

[22] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *CVPR*, 2021. 3

[23] Jiaman Li, Yihang Yin, H. Chu, Y. Zhou, Tingwu Wang, S. Fidler, and H. Li. Learning to generate diverse dance motions with transformer. *ArXiv*, abs/2008.08171, 2020. 2

[24] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef J Sivic. Estimating 3d motion and forces of person-object interactions from monocular video. In *CVPR*, 2019. 3

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 3

[26] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ToG*, 39(4):40–1, 2020. 1, 2

[27] C Karen Liu and Sumit Jain. A quick tutorial on multibody dynamics. *Online tutorial, June*, page 7, 2012. 4

[28] Libin Liu, KangKang Yin, and Baining Guo. Improving sampling-based motion control. In *Computer Graphics Forum*, volume 34, pages 415–423. Wiley Online Library, 2015. 2

[29] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. *ToG*, 29(4):128, 2010. 2

[30] Miao Liu, Dexin Yang, Yan Zhang, Zhaopeng Cui, James M Rehg, and Siyu Tang. 4d human body capture from egocentric video via 3d scene grounding. *arXiv preprint arXiv:2011.13341*, 2020. 3

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 34(6):248:1–248:16, 2015. 3

[32] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 1, 6

[33] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017. 2

[34] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect. *ToG*, 39(4), 2020. 3

[35] Igor Mordatch, Kendall Lowrey, Galen Andrew, Zoran Popovic, and Emanuel V. Todorov. Interactive control of diverse complex characters with neural networks. In *NeurIPS*, 2015. 2

[36] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *SIGGRAPH*, pages 137–144, 2012. 1, 2, 4

[37] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ToG*, 31(4):1–8, 2012. 1, 2

[38] Igor Mordatch, Jack M Wang, Emanuel Todorov, and Vladlen Koltun. Animating human lower limbs using contact-invariant optimization. In *Siggraph Asia*, 2013. 4

[39] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006. 5

[40] Zherong Pan, Bo Ren, and Dinesh Manocha. Gpu-based contact-aware trajectory optimization using a smooth force model. In *SIGGRAPH*, 2019. 2

[41] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ToG*, 37(4):1–14, 2018. 2

[42] Paul S. A. Reitsma and Nancy S. Pollard. Evaluating motion graphs for character animation. *ToG*, 26(4), 2007. 2

[43] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *ECCV*, 2020. 2, 3, 4, 5

[44] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ToG*, 39(6), 2020. 2, 3, 4, 5, 6, 7

[45] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1):4–27, 2010. 5

[46] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ToG*, 38(6):209–1, 2019. 2

[47] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ToG*, 39(4), 2020. 2

[48] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3

[49] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 3

[50] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *CVPR*, 2008. 3

[51] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion. *arXiv preprint arXiv:2011.15119*, 2020. 2

[52] Alexander W Winkler, C Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018. 2

[53] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ToG*, 39(4):33–1, 2020. 2

[54] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *ECCV*, 2018. 2

[55] Ze Yang, Siva Manivasagam, Ming Liang, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Recovering and simulating pedestrians in the wild. *arXiv preprint arXiv:2011.08106*, 2020. 1

[56] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *ECCV*, 2020. 2, 5, 8

[57] Ye Yuan and Kris Kitani. Residual force control for agile human behavior imitation and extended motion synthesis. In *NeurIPs*, 2020. 2

[58] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *ICCV*, 2019. 2

[59] Yan Zhang, Michael J. Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *CVPR*, 2021. 1, 2, 3

[60] Tiancheng Zhi, Christoph Lassner, Tony Tung, Carsten Stoll, Srinivasa G. Narasimhan, and Minh Vo. Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020. 3

[61] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *ICLR*, 2018. 2